

৩টা জিনিস পাবেন

১) সামান্য আর্কিটেক্ট সম্পর্কে ধারণা।

২) সমস্যা সমাধান সম্পর্কে কিছুটা ধারণা হবে।

৩)

Workflow Structure (Nodes, Triggers & Connections)

Deep_Research_Report_Generation (Work Flow)

এটা একটা Automation Workflow যা রিসার্চ/তথ্য সংগ্রহ থেকে শুরু করে রিপোর্ট তৈরি এবং সেটাকে Notion-এ সংরক্ষণ পর্যন্ত সবকিছু অটোমেটিক করে দেয়।

কাজের ধাপগুলো সহজ ভাষায়:

1. Trigger (শুরু)

- Workflow টা চালু হয় যখন কেউ Telegram-এ মেসেজ পাঠায় বা Webhook থেকে ডাটা আসে।

2. Strategy Agent (রিসার্চ প্ল্যানিং)

- ইউজারকে জিজ্ঞেস করে কোন টপিক নিয়ে রিসার্চ করতে হবে।
- প্রয়োজনে ইউজারের থেকে আরও ক্লিয়ার ইনফো নেয়।

3. Query Generation (সার্চ কিওয়ার্ড বানানো)

- যে টপিক দেয়া হলো তার উপর ভিত্তি করে সার্চ ইঞ্জিনে ব্যবহার করার মতো কিওয়ার্ড/কুয়েরি বানায়।

4. Web Search (তথ্য খোঁজা)

- Tavily API ব্যবহার করে গুগলের মতো সার্চ রেজাল্ট আনে।

5. Content Extraction (কন্টেন্ট বের করা)

- সবচেয়ে relevant ওয়েবসাইটগুলো থেকে দরকারি কন্টেন্ট তুলে আনে।

6. Report Generation (রিপোর্ট বানানো)

- AI (OpenAI) দিয়ে সংগৃহীত ডাটা থেকে সুন্দর করে Markdown ফরম্যাটে রিপোর্ট বানায়।

7. Notion Integration (রিপোর্ট সংরক্ষণ)

- তৈরি হওয়া রিপোর্ট আর মেটাডাটা সরাসরি আপনার Notion Database-এ সেভ করে।

8. Notification (খবর দেয়া)

- রিপোর্ট রেডি হলে ইউজারকে Telegram বা অন্য চ্যানেলে জানিয়ে দেয়।

multi-agent orchestration + human review)

🔑 মূল উদ্দেশ্য

1. ইউজার কোনো **Input Data** পাঠাবে (Webhook Trigger দিয়ে শুরু হয়)।
2. একটা **Meta-Orchestrator Agent (GPT-4o)** ঠিক করবে – কোন কোন এজেন্ট কী ক্রমে কাজ করবে (Summarizer, Synthesizer, Peer Reviewer, Sensemaking Agent, Prompt Engineer, Onboarding, Archivist ইত্যাদি)।
3. এজেন্টরা একে একে কাজ করবে, কন্টেন্ট তৈরি/রিভিউ করবে।
4. শেষে কন্টেন্ট **Archivist Agent** → human reviewer এর কাছে approval এর জন্য যাবে।
5. মানুষ approve করলে সেটা **Postgres DB** এবং (যদি enable থাকে) **GitHub Repo** তে save হবে।
6. Reject করলে feedback যাবে → system আবার content refine করে দিতে পারবে।

Meta-Orchestrator মূলত একাধিক অটোমেশন/এজেন্টকে একসাথে পরিচালনা করার মাস্টার কন্ট্রোল সিস্টেম বোঝায়।

সহজভাবে বললে-

- **Orchestrator** = একটি নির্দিষ্ট সেটের টাস্ক, এজেন্ট বা ওয়ার্কফ্লোকে সঠিক ক্রমে চালানো ও ম্যানেজ করা।
- **Meta-Orchestrator** = একাধিক Orchestrator (বা এজেন্ট নেটওয়ার্ক) একসাথে ম্যানেজ করার “সুপার লেয়ার”।

উদাহরণ

1. সাধারণ Orchestrator

- একটি AI এজেন্ট শুধু লিড জেনারেট করবে।
- আরেকটি AI এজেন্ট শুধু কন্টেন্ট লিখবে।
- আরেকটি AI এজেন্ট শুধু ইমেইল পাঠাবে।

2. Meta-Orchestrator

- এগুলোর মধ্যে কে আগে কাজ করবে, কে পরে কাজ করবে, কখন কোন এজেন্টের আউটপুট অন্য এজেন্টকে দেওয়া হবে – এগুলো নিয়ন্ত্রণ করবে।
- যেমন: লিড জেনারেশন এজেন্ট নতুন ডাটা আনলে → কন্টেন্ট এজেন্ট সেই ডাটার উপর ভিত্তি করে টেক্সট বানাবে → তারপর ইমেইল এজেন্ট পাঠাবে।

টেকনিক্যাল দৃষ্টিকোণ থেকে বললে-

- **Meta-Orchestrator = Multi-agent Orchestration Layer**
- কাজগুলো:
 - এজেন্ট/ওয়ার্কফ্লোদের মধ্যে **Coordination**
 - **Error Handling** (যদি কোনো এজেন্ট ব্যর্থ হয়, বিকল্প এজেন্ট চালানো)
 - **Prioritization** (কোন এজেন্ট আগে কাজ পাবে)
 - **Resource Optimization** (CPU, API call limit, data flow)

রিয়েল-লাইফ ব্যবহার-

- **AI Automation Company:** একই সাথে ১০+ ক্লায়েন্টের জন্য ভিন্ন ভিন্ন workflow চলছে। Meta-Orchestrator এগুলো নিয়ন্ত্রণ করবে।
- **AI Village বা Multi-Agent System:** যেখানে শত শত এজেন্ট বিভিন্ন কাজ করছে, সেখানে Meta-Orchestrator “মাস্টার ম্যানেজার” হিসেবে কাজ করবে।

CV Insight AI

একটি প্রার্থী (Candidate) যখন তার রেজুমে (CV / Resume) পাঠায়, সেটা PDF ফরম্যাটে থাকে। এই ওয়ার্কফ্লো PDF ফাইলটিকে ছবি (image) তে রূপান্তর করে, তারপর একটি **Vision Language Model (VLM)** ব্যবহার করে সেই রেজুমের তথ্য বিশ্লেষণ করে। শেষ পর্যন্ত, মডেল বলে দেয় যে প্রার্থীটি নির্দিষ্ট চাকরির জন্য উপযুক্ত কি না।

AI Personal Shopper + Store Assistant

“একজন গ্রাহক চ্যাটে প্রোডাক্ট বা স্টোর সংক্রান্ত প্রশ্ন করলে, AI Personal Shopper এজেন্ট তা বুঝে WooCommerce থেকে প্রোডাক্ট খুঁজে বা Google Drive/RAG সিস্টেম থেকে তথ্য নিয়ে তা উত্তর দেয়।”

Scenario:

গ্রাহক চ্যাটে লিখল: *“I want a red leather handbag under \$70.”*

→ এজেন্ট বুঝল এটা প্রোডাক্ট খোঁজার চাহিদা, WooCommerce থেকে matching হ্যান্ডব্যাগ খুঁজে AI রেসপন্স তৈরি করে গ্রাহককে দেখায়।

যদি গ্রাহক জিজ্ঞেস করে: *“When does the store open?”*

→ এজেন্ট বুঝল এটা প্রোডাক্ট নয়, RAG + Google Drive থেকে তথ্য নিয়ে উত্তর দেয়।

আপনি চাইলে আমি এটাকে আরও সুপার সংক্ষিপ্ত ১ লাইনে স্টুডেন্টদের জন্য catchy sentence বানিয়ে দিতে পারি।

1. Introduction (5 মিনিট)

- Workflow = Nodes + Triggers + Connections
- n8n এ প্রতিটি automation workflow এই তিনটার উপর ভিত্তি করে গড়ে ওঠে।

2. Nodes (15 মিনিট)

Definition: Node মানে একেকটা কাজ বা action।

অফিসিয়ালভাবে ~১,১৩৯ integrations আছে; community-published nodes প্রায় ২,০০০; মোট মিলে ~৩,০০০+ পর্যন্ত এক্সটেনশন হতে পারে (বর্তমান snapshot)।

<https://n8n.io/integrations>

📌 Node এর ধরন:

- **Trigger Node** → Workflow শুরু করার জন্য (যেমন Webhook, Cron, Schedule, Form trigger, App event trigger)

এগুলোকে broadly ৪ ধরনের গ্রুপে ভাগ করা যায়:

1. **Time-based Triggers** → সময় অনুযায়ী workflow শুরু করে

- **Cron Trigger** (সুনির্দিষ্ট সময়/প্যাটার্নে)
- **Schedule Trigger** (একবার বা নির্দিষ্ট interval-এ)
- 2. **Webhook/Event-based Triggers** → বাইরের কোনো API/সার্ভিস থেকে event এলে ট্রিগার হয়
 - **Webhook Trigger**
 - **App/Webhook-based triggers** (যেমন: Stripe Trigger → payment success event, Slack Trigger → message posted ইত্যাদি)
- 3. **Form/User Input Triggers**
 - **Form Trigger** (n8n form submission)
 - **Manual Trigger** (workflow ডিবাগ বা টেস্টের সময় হাতে চালানো হয়)
- 4. **Application Specific Triggers (OAuth2/Integration apps)**
 - যেমন: Gmail Trigger, Notion Trigger, Trello Trigger, Airtable Trigger, Discord Trigger ইত্যাদি — এগুলো সব অ্যাপের event শুনে workflow চালায়।
- **Action Node** → কোনো কাজ execute করার জন্য (যেমন Gmail → Send Email, Airtable → Add Record)
- **Data Node** → ডাটা পরিবর্তন বা ফিল্টার করার জন্য (যেমন IF, Merge, SplitInBatches, Set, Function node)

3. Triggers (15 মিনিট)

Definition: Trigger হল workflow শুরুর জন্য দরকারি “event”.

☞ Trigger এর ধরন:

1. **Webhook Trigger** → বাইরের সিস্টেম থেকে request এলে workflow চলবে।
 - Example: একটি ফর্ম সাবমিট হলে workflow চালু।
2. **Schedule Trigger (Cron)** → নির্দিষ্ট সময় বা interval এ workflow চালু।

- Example: প্রতিদিন সকাল ১০টায় report পাঠানো।
- 3. **App-based Trigger** → যেমন Gmail New Email, Airtable New Record, Slack New Message।

Demo Idea:

- একটি Cron Trigger দিন → প্রতি মিনিটে workflow চালু হবে।
- Output এ “timestamp” log করে দেখান।

4. Connections (10 মিনিট)

Definition: Node → Node data flow।

☞ Types of Connections:

- **Direct Connection** → এক node এর output → অন্য node এর input।
- **Conditional Connection** → IF Node দিয়ে true/false branch।
- **Merge/Combine Connection** → একাধিক ডাটা একত্রিত করা।